Pencil Sketch

http://www.askaswiss.com/2016/01/how-to-create-pencil-sketch-opencv-python.html

Using OpenCV and Python, an RGB color image can be converted into a pencil sketch in four simple steps:

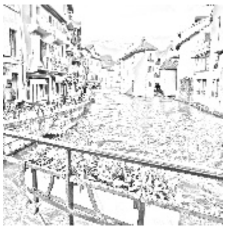
- Convert the RGB color image to grayscale.
- Invert the grayscale image to get a negative.
- Apply a Gaussian blur to the negative from step 2.
- Blend the grayscale image from step 1 with the blurred negative from step 3 using a color dodge.

Load the picture that will be sketched

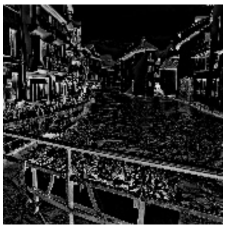
```
(ns opencv3.cartoon2
 (:require
  [opencv3.core :refer :all]
  [opencv3.utils :as u]))

(def img
  (-> "https://cdn.theculturetrip.com/wp-content/uploads/2016/01/canals2.jpg"
  u/mat-from-url
  (u/resize-by 0.10)))
(u/mat-view img)
```

```
(def gray (-> img clone (cvt-color! COLOR_BGR2GRAY)))
(u/mat-view gray)

(def inverted
  (-> gray clone (bitwise-not!)))
(u/mat-view inverted)

(def gaussed
  (-> inverted clone (gaussian-blur! (new-size 21 21) 0.0 0.0)))
(u/mat-view gaussed)
```

```
(defn dodge-v2! [img_ mask]
  (let [ output (clone img_) ]
  (divide img_ (bitwise-not! (-> mask clone)) output 256.0)
  output))

(u/mat-view (dodge-v2! gray gaussed))
```



```
(defn burn-v2! [ image mask]
  (bitwise-not! (dodge-v2! image mask)))
(u/mat-view (burn-v2! gray gaussed))
```

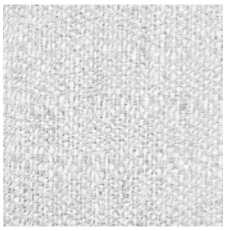


Apply a Canvas effect

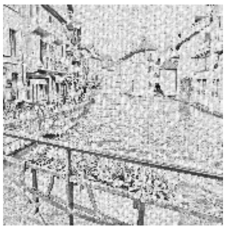
Now that the main picture has been turned to a crayon styled art form, it would be nice to lay this out on a canvas looking mat.

This is done using the **multiply** function from OpenCV core.

```
(def canvas (imread "resources/canvas.jpg" 0))
(resize! canvas (new-size (.cols gray) (.rows gray)))
(u/mat-view canvas)
```



```
(def output (new-mat))
(multiply (dodge-v2! gray gaussed) canvas output  (/ 1 256.0))
(u/mat-view output)
```

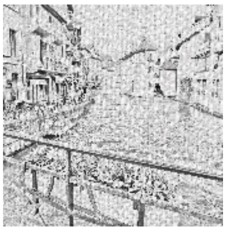


Let's make a function out of the above, so we can apply a few different canvas and see the output and effect of each of them.

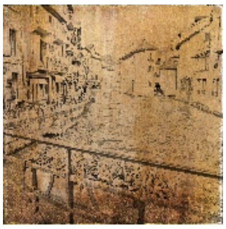
```
(defn apply-canvas! [ sketch canvas]
  (let [ output (new-mat)]
    (resize! canvas (new-size (.cols sketch) (.rows sketch)))
    (multiply (-> sketch clone (cvt-color! COLOR_GRAY2RGB)) canvas output  (/ 1 256.0))
    output  ))
```

```
#'opencv3.cartoon2/apply-canvas!
```

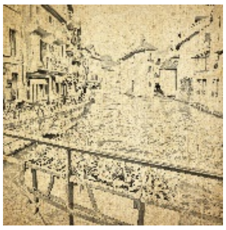
```
(def sketch (dodge-v2! gray gaussed))
(def canvas (imread "resources/canvas.jpg"))
(u/mat-view (apply-canvas! sketch canvas))
```



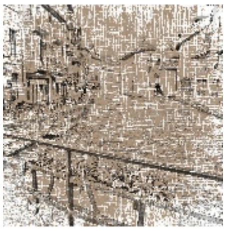
```
(def canvas (imread "resources/canvas/oldcanvas.jpg"))
(u/mat-view (apply-canvas! sketch canvas))
```



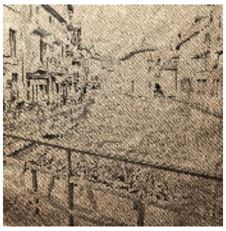
```
(def canvas (imread "resources/canvas/dottedcanvas.jpg"))
(u/mat-view (apply-canvas! sketch canvas))
```



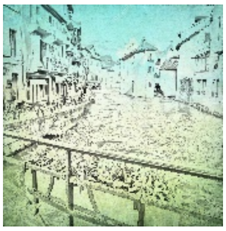
```
(u/mat-view (apply-canvas! sketch  (imread "resources/canvas/oldcanvastexture.jpg")))
```



```
(u/mat-view (apply-canvas! sketch (imread "resources/canvas/vintage-old-brown-canvas-texture.jpg")))
```



```
(u/mat-view (apply-canvas! sketch (imread "resources/canvas/grunge-parchment-background-canvas.jpg")))
```



```
(u/mat-view (apply-canvas! sketch (imread "resources/canvas/japanese_paper.jpg")))
```